



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

TRABAJO FINAL DEL

REALIZADO POR

TUTORIZADO POR

FECHA: Valencia,

AGRADECIMIENTOS

Tras finalizar el desarrollo de este trabajo fin de grado, me siento totalmente en el deber de agradecer a todas esas personas que me ayudaron tanto técnica como emocionalmente en el transcurso de este proyecto.

En primer lugar, me gustaría agradecerle a mi tutor Jorge Alcaide las horas dedicadas para ayudarme a gestionar y plantear este proyecto, así como las revisiones y sugerencias planteadas. Sin él nada de esto hubiese sido posible. Quisiera agradecerle que me dejara participar en este proyecto global sobre digitalización del museo de ciencias naturales de Valencia.

También quisiera darles las gracias a mis padres por su dedicación en potenciar mi rendimiento y recordarme que el esfuerzo y la perseverancia son los pilares para el trabajo bien hecho. También quisiera agradecer a mi hermano que me iniciara en todo este mundo de la tecnología virtual y me revelara el camino de las ingenierías y la técnica.

Por último, pero no por ello menos importante, no podría olvidarme de mis amigos, cuyo constante apoyo moral ha sido crucial no solo en el desarrollo de este trabajo sino de todo el grado de Diseño Industrial.

INDICE:

1. Objeto y objetivos	1
2. Inspiración y Referencias	2
3. Antecedentes	8
4. Concepto	9
5. Motores gráficos	10
6. Escaneado 3D.....	12
7. Exportar diseños 3D	14
8. Elementos gráficos	15
9. Funcionalidad.....	21
10. Descripción detallada de la solución adoptada.....	29
11. Alternativas propuestas	33
12. Conclusiones	34
13. Fuentes de Información:.....	35

1. Objeto y objetivos

Con el presente Trabajo Fin de Grado (TFG) se pretende diseñar una aplicación para smartphones que pueda ejercer de vínculo entre los visitantes del museo de Ciencias Naturales de Valencia y los fósiles expuestos.

El objetivo principal de este trabajo es romper con los estándares de las visitas a los museos diseñando una interacción entre el visitante y el entorno que visita, y que a través de la interacción y la estimulación táctil aumente la sensación de conectividad con los elementos que le rodean. Es bien conocida la voluntad de cualquier visitante en un museo de poder tocar los fósiles que ve, pero esto está lógicamente prohibido por la conservación de estos. Es por este motivo por el que se elabora esta aplicación, en la que el usuario puede a través de su teléfono tocar y ver de cerca aquellas piezas expuestas detrás de las vitrinas. De esta forma se crea un acercamiento entre visitante y exposición mucho más directa. Este trabajo parte del desarrollo previo que se realizó escaneando en 3D los cráneos de los dinosaurios que componen la exposición de Rodrigo Botet en el museo de Ciencias Naturales de Valencia. El diseño y el desarrollo de la aplicación han sido a través del motor gráfico Unity 3D, ya que incorpora una multitud de herramientas que facilitan importar objetos escaneados a su espacio virtual para el correcto desarrollo de la aplicación. En este trabajo se ha estudiado como es el visitante medio de este tipo de museos y se han tenido en cuenta factores como la ergonomía digital, la facilidad visual y la policromía de fácil comprensión.

Para un trabajo como este se debe tener muy en cuenta factores como la forma de interacción que tienen los usuarios con las aplicaciones de este tipo, o como la sensación que provoca el poder interactuar con los fósiles de unas maneras concretas. Asimismo, como la posibilidad que la usen personas con poca visibilidad, que el museo está incentivando el factor del tacto para los visitantes y habilita zonas para que se puedan tocar meteoritos y huesos.

Otros objetivos secundarios para el alumno en relación con el desarrollo de este trabajo son:

- Aprendizaje de programa Unity 3D, ya que es el software líder en el uso para desarrollo virtual tanto en tres dimensiones como en 2D, cualidad que viene perfecta para la aplicación ya que pretende tener tanto unos menús digitales como la interacción con los cráneos digitalizados en tres dimensiones.
- Aprender el uso y manipulación de ficheros creados a partir de la digitalización de piezas como en este caso son los cráneos de dinosaurio. Entender la relación nivel de detalle y peso del archivo y optimizarla.
- Entender como el usuario interacciona con una aplicación de smartphone para desarrollar y perfeccionarlo para la comodidad y permitir un uso intuitivo.

2. Inspiración y Referencias

Para el diseño de una aplicación para smartphones, lo primero que se debería hacer es ver otras aplicaciones con temática parecida y aprender cómo se estructuran y que funciona en cada una de ellas. Pero antes, hay que recordar que se pretende con esta aplicación y por tanto imaginar cómo queremos que sea y así sabremos que referencias buscar para inspirarnos. Lo primero es establecer que tratamos con una aplicación sobre dinosaurios, de carácter educativo, que incorpora una temática natural o que inspire “naturaleza”. Hablamos de una aplicación pública, destinada a los visitantes de museos (que en su mayoría son comprendidos en dos grupos: niños de excursión y parejas mayores). Una vez establecido esto, podemos realizar una búsqueda de aplicaciones parecidas que cumplan estos o algunos de estos rangos.

Algunas de las recogidas son:

- **MAN Museo Arqueológico**



Ilustración 1 - Icono app MAN

Esta aplicación del Museo Arqueológico Nacional propone una estructura en 3 fases: El inicio con el menú, una zona intermedia donde seleccionar la parte que quieres explorar y una parte final que ya te permite el uso de esa instancia. La temática visual se reduce a letras blancas simples sobre fondo blanco apoyados con formas simples como círculos e imágenes.



Ilustración 2 - Menú MAN

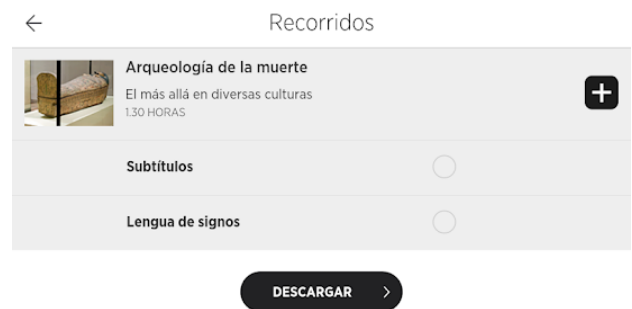


Ilustración 3 - Selector MAN



Ilustración 4 - Contenido MAN

- **Google Arts & Culture**



Ilustración 5 - Icono app
Google Arts & Culture

Esta aplicación de Google enfocada a un contenido más general está enfocada de manera que en la primera pantalla que ves, ya puedes elegir que contenido quieres explorar. Esto es debido a que no está especializada en un solo museo o exposición, si no que lo hace con un recopilatorio de muchas partes de multitud de museos a nivel mundial. Cuando entramos en el contenido, la forma que tiene de proponer una interacción del usuario con ese tema es a partir de recursos gráficos como imágenes, diagramas o videos.

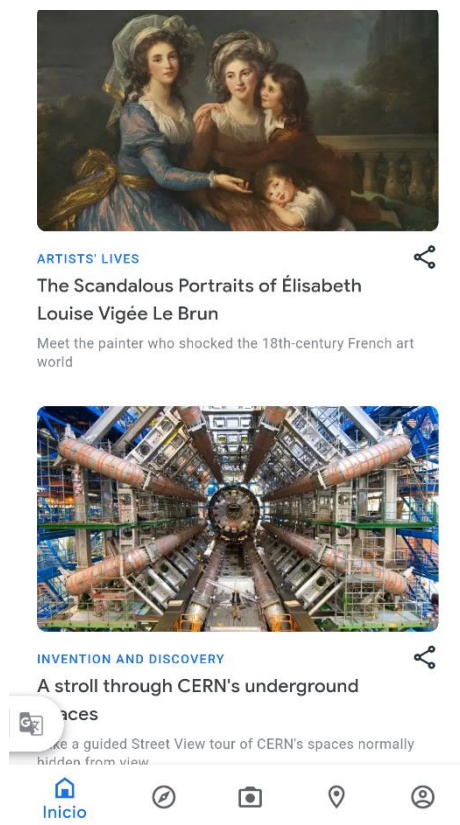


Ilustración 6 - Selección Google

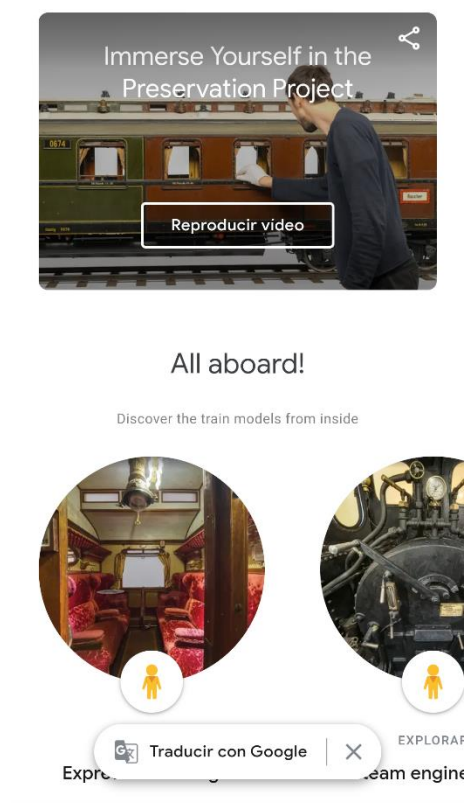


Ilustración 7 - Contenido Google

- Museo de Altamira



Ilustración 8 - Icono app Museo Altamira

Esta aplicación del Museo de Altamira comienza con un menú muy sencillo para seleccionar el idioma donde solo encontramos el título de la aplicación y una imagen de fondo. Posteriormente, pasamos a una pantalla selectora de contenidos, que a su vez nos lleva a otra pantalla para elegir un subtema del género escogido. Finalmente, el contenido se realiza de forma audiovisual, pero con las opciones de subtítulos, lenguaje de signos o audio descripción.



Ilustración 9 - Inicio Altamira



Ilustración 10 - Selector 1 de Altamira



Ilustración 12 - Opciones para el contenido Altamira



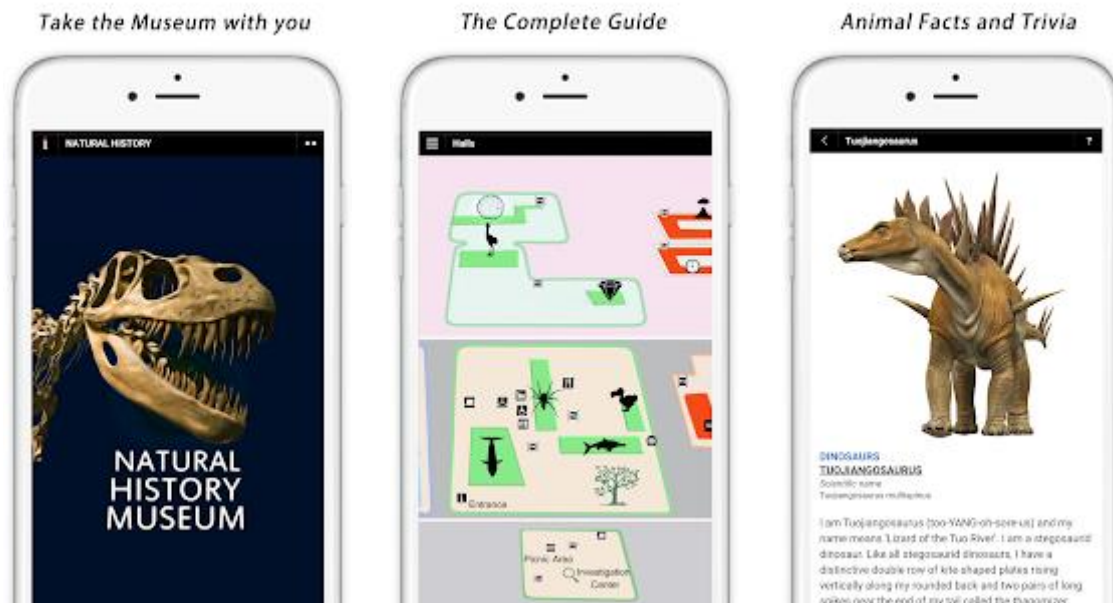
Ilustración 11 - Contenido Altamira

- **Natural History Museum**



*Ilustración 13 - Icono app
Natural History Museum*

Esta aplicación del Museo de historia natural de Londres ofrece una estructura diferente a lo que hemos visto hasta ahora. Se estructura en diferentes partes: Primero, el menú, es permanente en la parte inferior en vez de estar solo en la primera pantalla. En el podemos entrar a escoger el contenido que queremos explorar del museo, lo que nos lleva a otra pantalla selectora con los dinosaurios que podemos encontrarnos allí. Cuando seleccionamos uno, ya podemos ver imágenes y información a modo de texto que nos aporta información adicional. La interacción del usuario con lo que explora en su teléfono móvil es poca, ya que en esta aplicación solo se pretende acercare el museo a aquellos que lo visitaron en el pasado o quienes aún no lo han hecho, a diferencia de nuestro objetivo, que son los visitantes actuales del propio museo. Otra cosa por lo que destaca esta aplicación es la forma de presentar las opciones de selección al usuario, en un mapa con todos los símbolos de las distintas exposiciones y donde el usuario puede pulsar para adentrarse en ella.



*Ilustración 14 - Inicio y menú
Museo Londres*

*Ilustración 15 - Pantalla selectora
museo Londres*

*Ilustración 16 - Contenido museo
Londres*



Esta aplicación del Museo de Historia Natural de Frankfurt plantea una estrategia diferente, donde es el usuario quien se puede organizar sus exploraciones. Empieza con un menú muy escueto donde da la opción de hacer los tours. Esto te lleva a una pantalla selectora en la que ves todas las opciones a la vez y puedes seleccionar aquellas que deseas explorar. Posteriormente, la interacción de basa en presentar la sección donde se encuentra esta parte de exposición y posteriormente enseñar imágenes del dinosaurio en si acompañado con texto informativo y una opción de audio. Una de las principales prestaciones de esta aplicación es que te permite configurar un orden en la presentación de los contenidos y hacer un tour todo seguido sin tener que entrar en una cosa, salir, entrar en otra, salir otra vez... La continuidad y la correcta policromía hacer que la aplicación cobre una integridad visual muy importante.



Ilustración 18 - Menú museo Frankfurt



Ilustración 19 - Pantalla selectora Museo Frankfurt



Ilustración 20 - Contenido museo Frankfurt



Ilustración 21 - Mapa museo Frankfurt

Conclusiones de las referencias:

Debemos identificar las partes de cada aplicación que creemos que están mejor desarrolladas y identificar por qué gustan más y funcionan mejor. Por ejemplo, si empezamos por MAN (Museo Arqueológico Nacional), podemos destacar el buen uso de la congenialidad visual, es decir, una acertada paleta cromática, haciendo usos de colores opuestos (blanco para las letras, negro para fondos) además de un refuerzo positivo con geometrías sencillas como círculos y cilindros. Posteriormente, el uso de una simpleza y minimalismo ayudan mucho al usuario a saber lo que está haciendo. Si se sobreexpone al usuario a demasiada información a la vez se confunde y no es agradable. Una pantalla con pocas opciones y claras donde el usuario sepa que está haciendo en cada momento parece ser que es clave para un correcto uso de la aplicación. Si por ejemplo lo comparamos con la pantalla de selección de la app de Google, podemos ver como ellos también usan imágenes grandes con títulos explícitos para que se sepa qué es cada opción con claridad. Pero aquí vemos como para ir de opción en opción tenemos que hacer un desplazamiento vertical que cuando llevamos más de 6 o 7 opciones, ya no nos acordamos de las primeras, y de hacerlo, volver a ellas es demasiado camino. Quizás un sistema en rueda que enlace la última opción con la primera sea mejor elección. Google, no usa ningún estilo gráfico, lo que facilita un entendimiento visual, pero conecta poco con el usuario, ya que una temática digital en los botones, el fondo y las opciones, hace pensar al usuario que está sumergido desde el primer momento en el mundo que está explorando.

Por otra parte, la aplicación que presenta el museo de Altamira, engloba todo su contenido bajo un mismo tono ocre que nos recuerda perfectamente a las pinturas que nos enseña luego en su contenido, por lo que el uso correcto de la policromía parece que se debe tener muy en cuenta para favorecer al usuario la inmersión en el contenido. La selección del mismo, en esta opción se distribuye a través de imágenes que se desplazan de manera horizontal. Pero si debemos destacar algo de esta versión, es la posibilidad de activar las opciones concretas que el usuario desea como los subtítulos o el audio descripción.

Si vemos la aplicación de Natural History Museum de Londres, se asimila mucho al modelo de las otras aplicaciones, pero esta se centra en presentar una selección de contenido a través de un mapa digital donde el usuario identifica y escoge la parte del museo que quiere visitar. La parte en la que el usuario interactúa con el contenido se limita a unas imágenes del dinosaurio en cuestión acompañado de unos textos informativos y una pequeña sección de preguntas.

La última propuesta, la del museo de Frankfurt, es quizás, la más completa, propone un elaboradísimo sistema de selección que permite al usuario diseñar su propia "guía" y experimentar lo que él desea. Esto, acompañado de un resultante menú superior continuo en el que en cualquier momento podemos salir de la presentación y cambiar algún ajuste, se establece como una de las aplicaciones más trabajadas y que más les gusta a los usuarios (por su valoración el Google Play).

3. Antecedentes

En el campo de la interacción con una pieza de museo a través de un smartphone, no hay anteriores aplicaciones que permitan algo así, pues esta idea de TFG parte de la ausencia de este campo en el ámbito de las aplicaciones de museo. En cambio, si que podemos ver los objetivos que pretendemos para esta aplicación, como rotar un objeto o hacer zoom, en otras apps, como en diferentes juegos o simuladores. Estos son los principales objetivos a cumplir que hemos observado:

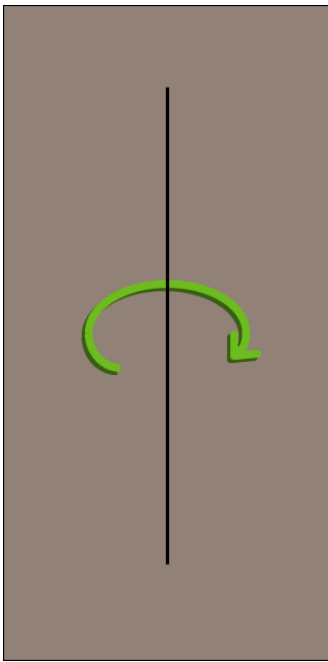


Ilustración 22 - Giro vertical

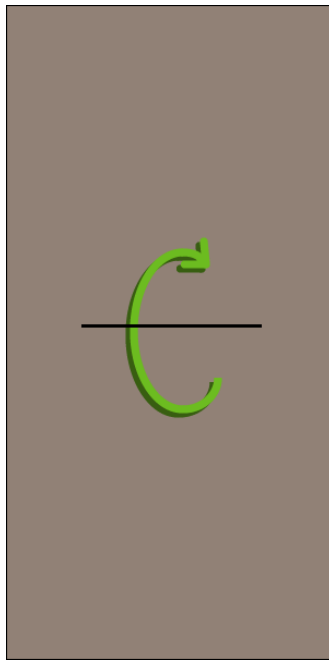


Ilustración 23 - Giro Horizontal

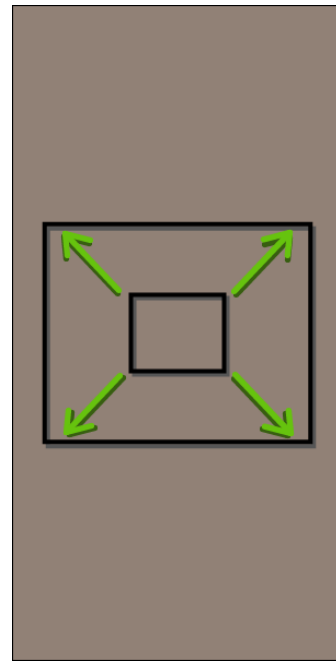


Ilustración 24 - Zoom

Para realizar estos movimientos a los cráneos del museo en nuestro smartphone, habrá que programar las ordenes de comandamiento para que los dedos en la pantalla táctil puedan ejercer esas ordenes al cráneo y controlar así el mismo. Estas órdenes están comúnmente estandarizadas con solo el uso de un dedo (índice) para los giros rotacionales y con el llamado "Pinch Zoom" para hacer el zoom, que consiste en tocar con dos dedos a la vez la pantalla (índice y pulgar) y acercarlos o alejarlos entre si para regular la distancia de la cámara con la que se ven los fósiles.

4. Concepto

En un trabajo como este, hay que formalizar muy bien el concepto a desarrollar, pues es el germen de lo que acabará siendo la propuesta final. Es bien conocido por cualquier diseñador industrial que el concepto o la idea de como se va a diseñar o desarrollar una cosa parte de las ideas generadas tras el análisis de las referencias que previamente se buscaron y desfragmentaron al detalle. Es con esta información, con la que se empieza a construir un concepto a desarrollar y con el que trabajar.

Para este trabajo, se parten de las siguientes premisas, recogidas de las conclusiones anteriores:

- Tres partes o pantallas diferenciadas, Menú, selección y contenido.
- Menú y pantalla de selección simples, para no confundir al usuario, ya que el objetivo de esta aplicación es muy concreto, no como el de las otras aplicaciones.
- Tonalidades cromáticas que ayuden al usuario a una inmersión temática.
- Primar la intuitividad del uso de la aplicación, tanto para el manejo de los menús como interactuando con el fósil.
- Añadir el uso de imágenes que afiancen el entendimiento para el usuario de como es con lo que está interactuando.
- Presentar un sistema de selección en el que se encadenen las opciones y en la última se pase a la primera, evitando así el largo desplazamiento entre unas opciones y otras.
- Evitar un texto informativo que distraiga al usuario de interactuar con el fósil, si no que sea una opción paralela que se pueda habilitar cuando se desea.
- Habilitar una opción para activar el audio descripción del texto informativo para la gente que le incomode leer mucho en smartphones.

5. Motores gráficos

Establecidas las directrices del proyecto, es momento de pensar en herramientas, y una de las más necesarias en este caso va a ser un software que nos permita trabajar en entornos virtuales 2D y 3D para importar nuestros ficheros con los cráneos escaneados y a su vez elaborar unos menús programables que nos permitan fácilmente guiar al usuario hasta la pantalla de interacción con los fósiles. Este tipo de programas son llamados hoy en día motores gráficos. En ellos se dispone de un espacio virtual, un punto fijo desde donde se observa, y se añaden los elementos en el escenario para configurar lo que queremos que ocurra y/o se vea.

Un motor gráfico es un software usado por aplicaciones y programas para dibujar gráficos en la pantalla de nuestro ordenador, smartphone o tablet. Desde hace unos años, la palabra motor en el mundo del software se usa de una manera diferente, refiriéndonos al software que ejecuta un determinado tipo de tareas comunes a muchas aplicaciones de software: un motor de base de datos, un motor de transcripción texto a escrito, o un motor gráfico. Más concretamente, se define como motor gráfico al framework de software diseñado para crear y desarrollar videojuegos o entornos digitales. Todo motor gráfico ha de ofrecer al programador una funcionalidad básica, proporcionando normalmente un motor de renderizado (“render”) para gráficos 2D y 3D. Los componentes que posee un motor de videojuego en la actualidad poseen una gran variedad de características muy diferentes entre sí. En principio, pueden distinguirse cinco grandes áreas:

- **La renderización:** es el proceso en el cual se generan los gráficos por ordenador a fin de mostrar en pantalla el aspecto visual del entorno virtual.
- **Polígonos:** Todo elemento que tenga una composición tridimensional es clasificado por polígonos. Es el procedimiento más primitivo en la creación de un mundo tridimensional y su complejidad geométrica varía de acuerdo con las capacidades técnicas del hardware.






Imagen					
Vértices	~5500	~2880	~1580	~670	140
Notas	Detalle Máximo. Jugador cerca del objeto.				Detalle Mínimo. Jugador muy lejos del objeto.

Ilustración 25 - Detalle de objetos según Wikipedia

- **Iluminación:** es la forma que el software tiene de trazar rayos de luz y reflejarlos uno a uno para crear sensaciones lumínicas reales dentro del entorno.
- **Lenguaje:** Todo software de entornos virtuales de renderización también necesita una parte de programación adjunta que le permita al desarrollador generar las ordenes y las acciones que tienen lugar dentro de el entrono virtual una vez se ejecuta la aplicación. Los más comunes son C++, C# y Java.
- **Licencia:** Como hay una gran variedad de programas de este tipo, algunos pueden incorporar una licencia de pago. Para este trabajo, usaremos aquellos que son gratuitos o solo basta con un registro.

Los motores gráficos gratuitos más populares son: 3D Rad, Source 2, Unreal Engine 4 y Unity 3D 5.0. La elección para desarrollar este proyecto es **UNITY3D**, ya que presenta algunas ventajas frente a los demás. Es muy popular, y por lo tanto en la red hay multitud de tutoriales para las cosas que queremos aprender en este trabajo y ponerlas en práctica. También destaca en la facilidad para importar ficheros 3D (como los cráneos que hemos escaneado) de una manera sencilla y rápida. Hasta hace poco la versión gratuita tenía ciertas limitaciones que se eliminaban pasando por caja, pero con Unity 5 se han eliminado todas esas restricciones. Esto quiere decir que la versión gratuita de este motor gráfico vendrá igual de completa que la versión de pago y la podemos aprovechar al máximo.

El Editor de Unity presenta herramientas múltiples que permiten una edición e iteración rápidas en tus ciclos de desarrollo, lo que incluye el modo Play para tener vistas previas rápidas de tu trabajo en tiempo real. Unity puede usarse junto con Blender, 3ds Max, Maya, Cinema 4D, Adobe Photoshop y un montón de formatos estándar en el tratamiento de objetos 3D como es nuestro caso con los cráneos. Los cambios realizados a los objetos creados con estos productos se actualizan automáticamente en todas las instancias de ese objeto durante todo el proyecto sin necesidad de volver a importar manualmente.

Unity, además, incluye la Asset Store, un recurso dentro del propio software donde los usuarios pueden descargar miles de paquetes de Assets en una amplia gama de categorías, incluyendo modelos 3D, texturas y materiales...



Ilustración 26 - Logo Unity

6. Escaneado 3D

Este proyecto parte de el trabajo elaborado por alumnos en TFG anteriores, que también se relacionan con el museo de Ciencias Naturales de Valencia. En concreto, estos alumnos organizaron la tarea de poder escanear en 3D los cráneos de los dinosaurios expuestos en la exposición permanente Rodrigo Botet. Para ello, trasladaron los cráneos hasta el estudio de la universidad politécnica de Valencia. Para realizar esta tarea se necesitan varios periféricos junto a un ordenador y un software para poder crear la malla del objeto correctamente. Esta clase de software se basan en la fotogrametría, o la superposición de muchas fotografías para dotar de dimensión a un objeto.

Lo primero que necesitamos es un escáner, en este caso, la universidad usa el HP Pro SLS-3 (ver imagen). Como podemos ver en su hoja de especificaciones del propio fabricante, podemos observar que es un dispositivo ideal para el escaneado digital de fósiles y huesos prehistóricos.

HP 3D Structured Light Scanner Pro S3 with Dual Camera Enables Archaeologists to Accurately Measure and Create Models of Prehistoric Artifacts



Deploying the HP 3D Scanner into the prehistoric artifact analysis workflow enables archaeologists to measure artifacts in detail and share the digital data with the accuracy, speed, and reliability they have never experienced before

Ilustración 27 - Titular HP sobre escáner 3D



Ilustración 28 - Escáner 3D HP Pro SLS-3

Este escáner, realiza miles de fotogramas para fotodetectar las caras del objeto que se escanea y así crear y posicionar los polígonos que compondrán la malla que resultará en el objeto 3D que solemos tratar como .OBJ (object). Una vez tenemos este fichero primigenio, se debe hacer un tratamiento digital para armonizar el mallado y resolver pequeños problemas de las partes más difíciles del objeto. Para ello se puede usar programas como el Meshmixer, que con sus herramientas nos permite en pocos pasos adecuar el resultante del escaneado para que ya lo podamos usar en nuestro proyecto. Podemos ver un ejemplo a continuación, donde se realiza un supervisado del mallado, un cerrado de agujeros y un calibrado del número de polígonos que usa el mallado, dependiendo de la finalidad del objeto.



Ilustración 29 - Logo Meshmixer

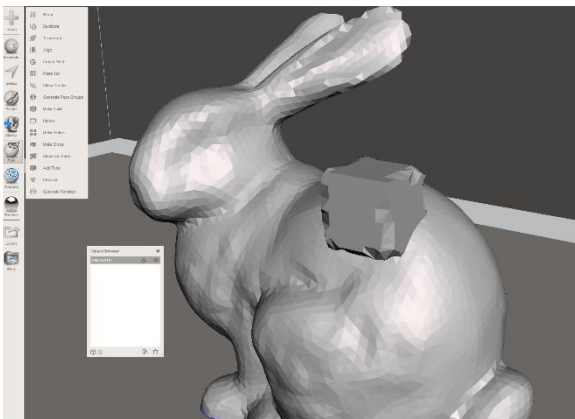


Ilustración 30 - Arreglado de agujeros en malla

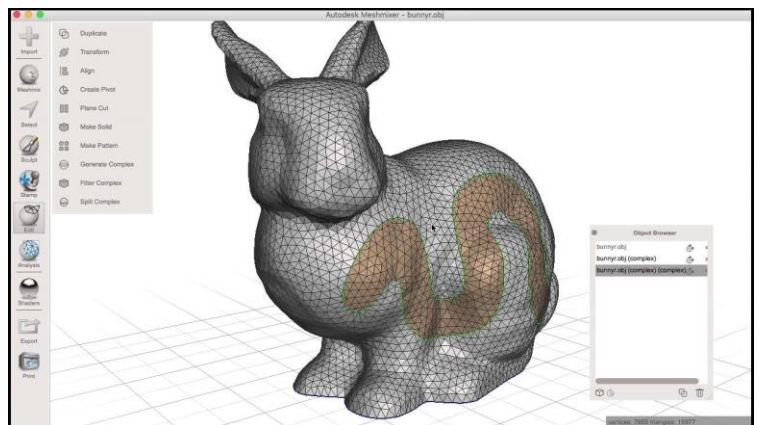


Ilustración 31 - Comprobación de mallado

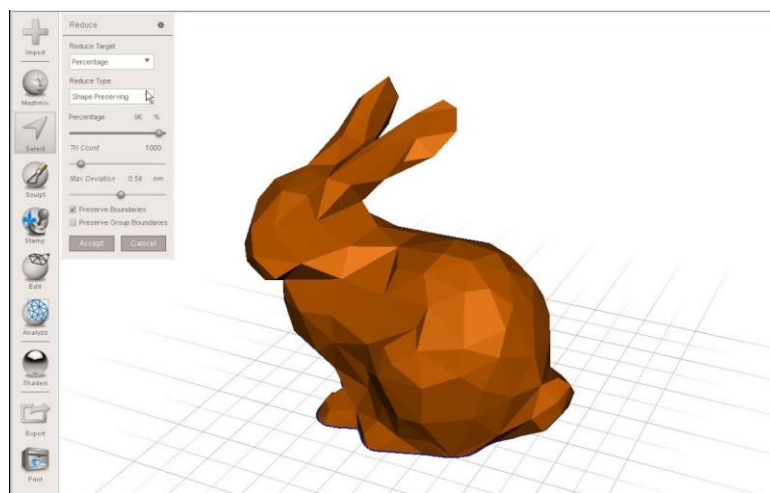


Ilustración 32 - Ajuste de polígonos

7. Exportar diseños 3D

Una vez tenemos los archivos de los objetos 3D digitales (.Obj), podemos pensar en exportarlos al motor gráfico con el que trabajamos, en nuestro caso UNITY 3D. Para ello, iniciamos un proyecto en la plataforma, y basta con copiar los ficheros de los cráneos en la ruta del mismo. Una vez hecho esto, nos aparecerán en la parte inferior del programa y podremos colocarlos en el entorno virtual. Aquí vemos los primeros problemas, como, por ejemplo, que el origen de coordenadas del objeto no está cerca del centro geométrico del objeto, lo que ocasiona rotaciones en ejes que no queremos. O también que los archivos pesan mucho y son muy pesados para los dispositivos con los que vamos a trabajar (smartphones), ya que no todo el mundo tiene teléfonos de alta gama. Para ello debemos utilizar un software de edición 3D especializado en estas cosas como,



Ilustración 33 - Icono Blender

por ejemplo, BLENDER. Blender es un programa informático multi plataforma, dedicado especialmente al modelado, iluminación, renderizado, animación y creación de gráficos tridimensionales. Aquí podremos

resolver cómodamente todas estas irregularidades que afectan a nuestros objetos virtuales. Blender se desarrolla como Software Libre, con el código fuente disponible bajo la licencia GNU GPL, su descarga y su uso es completamente gratuito. Una vez solucionados estos problemas, podemos volver a nuestro proyecto de Unity 3D y continuar con el proyecto.

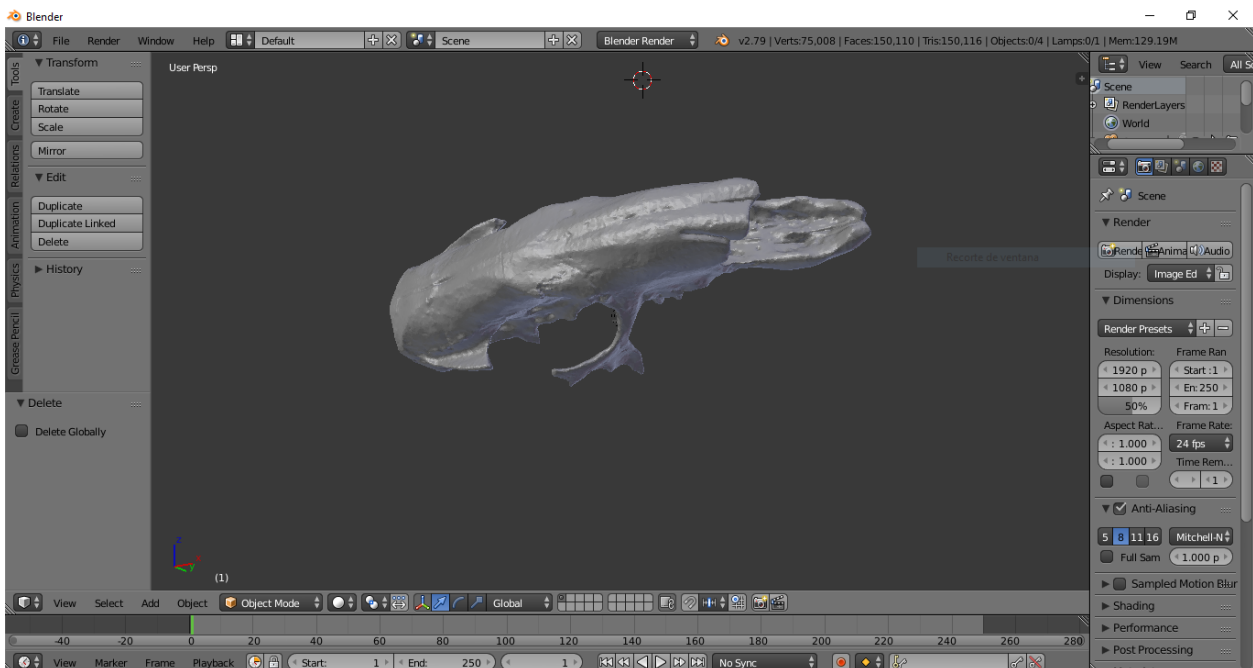


Ilustración 34 - Entorno Blender

8. Elementos gráficos

Los elementos gráficos, no solo son necesarios para la interacción del usuario con la aplicación, si no que suponen un gran refuerzo visual y ayuda inmersiva para el usuario que ayuda a conformar un entorno o un ambiente globalizado en el que quien la aplicación se sienta en comunión con lo que está explorando. Esto lo podemos ejemplificar muy bien con el ejemplo visto anteriormente de la aplicación del museo de Altamira, la forma de usar los colores ocres hace creer al usuario que ya está dentro de una cueva con pinturas. Por ello, es necesario hacer una reflexión de cual debe ser la paleta cromática que debemos usar en este proyecto. Teniendo en cuenta que todo el trabajo se desarrolla con el objetivo de interactuar con fósiles de dinosaurio, deberíamos sugerir unas tonalidades que nos recuerden lo natural, lo vivo, aquello salvaje en medio de la naturaleza. Hay un color que destaca por lo que la gente les atribuye respecto a estos temas, el verde, siempre estrechamente relacionado con la naturaleza y lo vivo. Por lo tanto, será este el color que usemos en nuestros botones y títulos.

A la hora de desarrollar una aplicación para móvil, sabemos que vamos a usar diferentes tipos de recursos digitales, como son los botones, los títulos, las flechas, los símbolos, las imágenes... Para ello vamos a ir paso a paso para ir construyendo nuestros elementos. Para esto, vamos a usar el software Illustrator de Adobe. No es gratuito, pero es el que nos han enseñado a usar en el grado, así que con su versión de prueba podemos trabajar bien. En orden de importancia, hablamos de:

- **Botones:** Los botones son pequeñas imágenes confeccionadas con un mismo estilo que resuelven con poco espacio la delimitación geométrica de donde tocar para ejercer una acción. Es importante que todos los botones se construyan igual, o mejor dicho, siguiendo el mismo patrón. Para ello vamos a explicar como se diseñaron las bases de un botón común cuyas características geométricas (largo y alto) se pudiesen cambiar fácilmente sin modificar la apariencia del mismo. Este botón que se ve a continuación se confecciona de un rectángulo cuyos vértices han sido totalmente redondeados. Posteriormente se le ha aplicado un sutil degradado entre las tonalidades verdes más

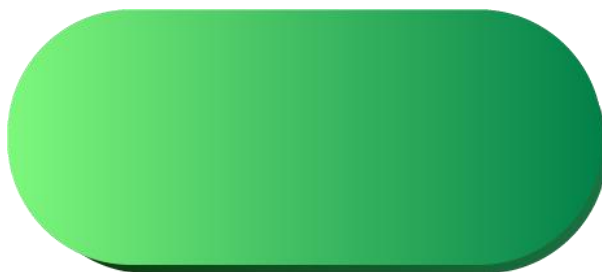


Ilustración 35 - Ejemplo de grafismo

claras y vivas a una más oscura para representar una luminosidad. Después de eso, se copia este rectángulo, se desplaza unos milímetros hacia abajo y hacia la derecha, se envía debajo del rectángulo primero y se modifica su degradado, haciéndolo inverso (de más oscuro a más claro) pero oscureciendo las dos partes. De esta forma hacemos una sensación de 3D para el botón.

- **Flechas:** Las flechas son muy necesarias para el momento de darle a entender al usuario que algo puede cambiar y que presione ahí para ejercer ese cambio en la selección de opciones. Para ello, se realiza un pequeño research (búsqueda) de cuales son los símbolos de flecha más usados en otras aplicaciones, y tras valorar la simpleza y fácil asimilación del usuario, se decide que se va a usar esta que ven debajo. Compuesta por dos líneas a 45° que se cruzan en su vértice, con una composición idéntica debajo para dar esa sensación de tridimensionalidad que hablábamos anteriormente. Éste es el resultado en un botón.

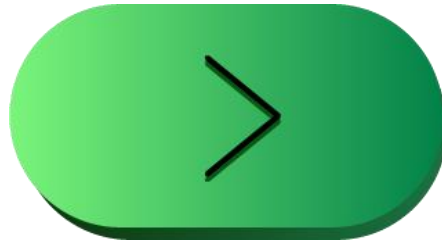


Ilustración 36 - Ejemplo de Flecha en un botón

- **Símbolos:** Los símbolos ayudan a entender cosas más rápidamente y son fáciles de identificar por los usuarios. Existen símbolos popularmente establecidos en la sociedad que hacen referencias a acciones que podemos usar dentro de nuestra aplicación, como, por ejemplo, “Salir”, “Entrar”, “Configuración” o “Información”. Para ello, con Illustrator se puede diseñar símbolos gráficos que sean fácilmente reconocibles y nos ayuden a establecer un menú y unas opciones más intuitivas. Los símbolos desarrollados son:

- o Entrar: Este símbolo genera una sensación en el usuario de pasar dentro, de acceder al contenido.

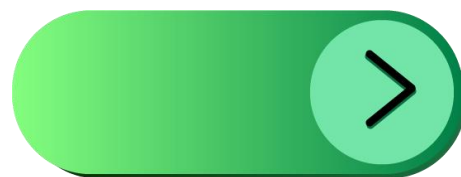


Ilustración 37 - Botón Entrar

- o Configuración Gráfica: El usuario entiende con un engranaje que se pueden modificar parámetros en su interior. Representa los ajustes que la aplicación deja que el usuario modifique.

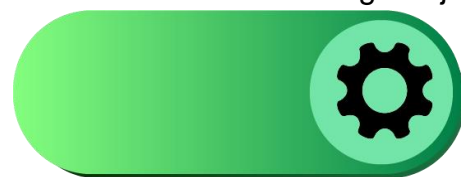


Ilustración 38 - Botón Configuración

- Información: Es muy popular en las ciudades las casetas de información para turistas, en ellas podemos encontrar este tipo de símbolos, sobre todo este de una “i” rodeada con un círculo. Este símbolo genera el entendimiento que dentro de esta opción puedes encontrar información básica sobre que es o que esperar de la aplicación.



Ilustración 39 - Botón Información

- Salir: Este símbolo es frecuentemente visto junto a los paneles de luz de emergencia en salas como cines o teatros. Es muy fácil de entender y ayuda al usuario a que sepa que en todo momento puede ejercer un sencillo toque para dejar esta aplicación.



Ilustración 40 - Botón Salida

- **Los títulos:** Para una aplicación de este estilo, unos buenos títulos den generar una sensación de inmersión para el usuario en el mundo que se está presentando. En este caso hablamos de el mundo jurásico, y por ello, debemos usar fuentes y colores que para el usuario sean convincentes y le transmitan la sensación que está visitando algo relacionado con dinosaurios. Para ello, y aunque suene un poco absurdo, debemos fijarnos en que fuentes son reconocibles para los usuarios y que las identifiquen con dinosaurios, y solo hay una, las que salen en las películas de “Jurassic Park”. Por lo que debemos usar una tipografía muy similar a esta. En concreto hemos encontrado: DOMINICAN.

DOMINICAN SMALL CAPS

Ilustración 41 - Fuente Dominican

Pero para el título de la aplicación, quizás es más adecuado una tipografía más austera, que no destaque y no acapare demasiado la atención del usuario, ya que queremos que se centre más en las opciones para acceder al contenido. Para esto usaremos la fuente “Champagne & Limousines”, que es más sutil y formal.

Champagne & Limousines

Ilustración 42 - Fuente Champagne & Limousines

- **Las imágenes:** Las imágenes que se usan son, o deberían ser, fieles representantes del contenido que se va a explorar, por ello, se decidió el uso de imágenes reales del propio museo para la selección de contenido. Lo podemos ver a continuación:



Ilustración 43 - Imagen Smilodon Ensenaednsis



Ilustración 44 - Glyptodont Munyizil



Ilustración 45 - Imagen Scelodotherium Carlesi



Ilustración 46 - Imagen Toxodon Byurmeisteri



Ilustración 47 - Imagen Mesotherium

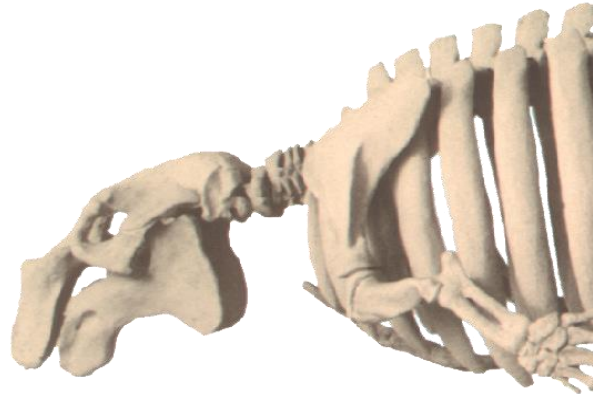


Ilustración 48 - Imagen Metaxytherium

Para el Metaxytherium, se prefirió usar una imagen creada por ordenador para darle mayor sensación de complejidad En el cuerpo.

Pero, para dentro del contenido, se pensó en que se podía disponer de los cráneos en contraposición de un fondo con una imagen con el dinosaurio vivo, para que el usuario relacionase la posición del cráneo en el cuerpo. Es por esto que se buscaron más imágenes que dispusieran una escena de realismo del ser para poder colocar enfrente el cráneo, esas imágenes son:



Ilustración 49 - Smilodón con vida



Ilustración 50 - Glytodont con vida



Ilustración 51 - Scelidotherium con vida



Ilustración 52 - Toxodon con vida



Ilustración 53 - Mesotherium con vida



Ilustración 54 - Metaxitherium con vida



Ilustración 55 - Composición cráneo - Dinosaurio

Para comprender el resultado de tener una imagen del dinosaurio en el fondo y el cráneo superpuesto, se añade una imagen de la composición para entender como quedaría el resultado y sentar la base para generar una correcta funcionalidad en el siguiente apartado.

Ahora podemos ver que lo que se pretende, es poder interactuar con el cráneo a partir de una posición que el usuario percibe como inicial, en el que se entiende la posición relativa de los cráneos y a partir de la cual, se empieza a trastocar para verla en detalle.

9. Funcionalidad

Para poder ejercer una correcta interacción con los fósiles, se pretende poder actuar una clase de movimientos que nos permitan apreciar en su totalidad los cráneos, por los 360° y en detalle. Para ello debemos programar cada uno de los movimientos que deseamos y asociarlo a una orden. Los movimientos que queremos imprimir sobre los cráneos, se explicaron anteriormente, pero los volvemos a recordar, eran: rotación en los 3 ejes y un zoom del fósil. Para esto debemos estructurarlo en dos partes.

Primero, en referencia a las rotaciones sin desplazamiento. La solución adoptada parte de un objeto invisible tridimensional, una esfera, la cual engloba en su interior el cráneo. De esta forma queremos que, para rotar el mismo, al tocarlo y desplazar el dedo, se ejerza un vector dirección que provee a el giro un sentido y un recorrido en el tiempo. Pero, los cráneos no son regulares ni continuos, por lo que su forma no es adecuada para esto. Es aquí donde entra la función de la esfera, la cual, aunque rote sobre si misma, presenta la misma forma sin irregularidades. Por ello se asocia el fósil en su interior y se fija de modo que al girar la esfera, se gira el cráneo.

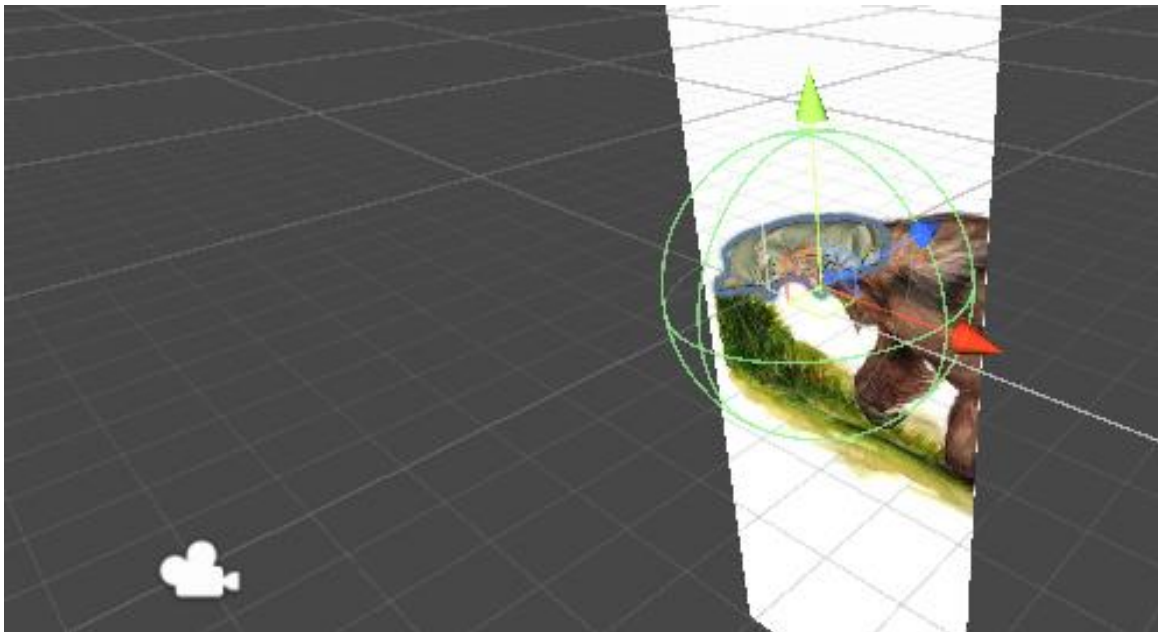


Ilustración 56 - Composición en Unity 3D

Aquí se puede observar como desde el punto de vista de la cámara, se ve la esfera, englobando el cráneo, delante de la imagen del propio dinosaurio. De esta forma, el centro geométrico del cráneo se ajusta al de la esfera y se produce una correlación entre rotaciones.

El siguiente paso es programar las ordenes para que cuando pulsemos la pantalla, giremos el cráneo. Para ello debemos tener muy claro como queremos que sean las ordenes que ejecutamos. Para este caso, solo queremos usar un dedo con el que pulsaremos la pantalla, y arrastraremos en una dirección. Con este movimiento, estamos estableciendo unos parámetros que podemos usar en

nuestro código. Primeramente, estamos pulsando la pantalla, lo que debemos detectar, además, de que lo hacemos solo con un dedo. Posteriormente, debemos detectar que estamos arrastrando el dedo por la pantalla, con lo que creamos un vector de dos dimensiones (del plano de la pantalla del dispositivo). Este vector, nos indica la dirección de la rotación y la continuidad de la rotación, es decir, desde que pulsamos, arrastramos y soltamos, se producen miles y miles de comprobaciones de donde estará el dedo en ese momento, y por lo tanto miles de vectores que se van actualizando. Esto le confiere a la rotación un tiempo definido para rotar. En cuanto no estemos incrementando el vector (arrastrando el dedo) o soltemos la pantalla, el fósil ya no rotará. Todo esto viene redactado en el código llamado “rotObj” (rotar objetos):

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class rotObj : MonoBehaviour
{
    float rotSpeed = 1;

    void OnMouseDown()
    {
        float rotX = Input.GetAxis("Mouse X") * rotSpeed * Mathf.Deg2Rad;
        float rotY = Input.GetAxis("Mouse Y") * rotSpeed * Mathf.Deg2Rad;

        transform.RotateAround(Vector3.up, -rotX);
        transform.RotateAround(Vector3.right, rotY);
    }
}
```

Primero vemos unas llamadas al software interno de Unity comunes en la mayoría de scripts (códigos). Luego podemos observar como se inicializa este código con “MonoBehaviour”. Dentro del cuerpo, lo primero en establecerse es una variable que hace referencia a la velocidad con la que rotaremos el objeto con respecto a los vectores que detectaremos, pero como queremos que se muevan con la misma velocidad, lo dejamos en 1 como valor definido. Luego iniciamos la función “OnMouseDown”, que viene con algunos factores predefinidos en Unity y nos ayudarán en nuestro trabajo. Definimos “rotX” y “rotY” que serán las rotaciones en cada uno de los ejes y vienen definidos por el vector que creamos al tocar y desplazar el dedo en la pantalla. Hecho esto, con “transform.RotateAround” le indicamos al objeto que se mueva, en este caso que rote, según esos vectores definidos anteriormente, en una mezcla de los ejes X e Y, haciendo que la combinación de estos pueda hacer que gire en 360°.

Para programar el zoo, debemos hacer una cosa parecida, pero pensando en los dos dedos. Para ejercer esta orden, tendremos que tocar con dos dedos en la pantalla, acercarlos o alejarlos, calcular la distancia y aplicar a la distancia de visión una diferencia de la distancia a la que se captan los objetos en la escena virtual. Veamos el código y expliquémoslo poco a poco.

```
using UnityEngine;

public class PinchZoom : MonoBehaviour
{
    public float perspectiveZoomSpeed = 20.0f;
    public float orthoZoomSpeed = 20.0f;

    void Update()
    {
        if (Input.touchCount == 2)
        {
            Touch touchZero = Input.GetTouch(0);
            Touch touchOne = Input.GetTouch(1);

            Vector2 touchZeroPrevPos = touchZero.position - touchZero.deltaPosition;
            Vector2 touchOnePrevPos = touchOne.position - touchOne.deltaPosition;

            float prevTouchDeltaMag = (touchZeroPrevPos - touchOnePrevPos).magnitude;
            float touchDeltaMag = (touchZero.position - touchOne.position).magnitude;

            float deltaMagnitudeDiff = prevTouchDeltaMag - touchDeltaMag;

            if (GetComponent<Camera>().orthographic)
            {
                GetComponent<Camera>().orthographicSize += deltaMagnitudeDiff * orthoZoomSpeed;

                GetComponent<Camera>().orthographicSize = Mathf.Max(GetComponent<Camera>().orthographicSize, 0.1f);
            }
            else
            {
                GetComponent<Camera>().fieldOfView += deltaMagnitudeDiff * perspectiveZoomSpeed;

                GetComponent<Camera>().fieldOfView = Mathf.Clamp(GetComponent<Camera>().fieldOfView, 25f, 60f);
            }
        }
    }
}
```

Primero establecemos las variables con las que delimitaremos la velocidad a la que se realiza el zoom respecto a la velocidad con la que movemos los dedos, estas son llamadas: "perspectiveZoomSpeed" y "orthoZoomSpeed", en referencia a si la visión de la escena se hace en perspectiva o en vista ortogonal.

Luego tenemos que establecer una pregunta, ¿Estamos tocando con dos dedos? Esto es necesario para activar la función y evitar errores con el código anterior. Esto se consigue con la clausula "IF" donde solo entraremos si se cumple la condición de que la pantalla detecta dos dedos pulsandola.

A partir de aquí se complica un poco, pero si vamos paso a paso es fácil de entender. Creamos un nombre para cada pulsación detectada, "Touch touchZero" y "Touch touchOne" que cojen la posición de las pulsaciones por un script interno de Unity llamado "Input.GetTouch(X)" donde equivale al vector posición del dedo X que ha pulsado.

Hecho esto, sabemos que vamos a mover los dedos, pero para ello antes hay que tener controlada la distancia en la que parten, y a la que llegan. Esto se descompone en miles de comprobaciones para ver si el dedo se desplaza y comprueba la distancia que estaban los dos dedos y a la que se han movido ahora. Con los vectores de dos dimensiones "touchZeroPrevPos" y "touchOnePrevPos" se crean dos variables con la posición donde estaban los dedos menos la que están ahora, creando así un diferencial de posición. Si este es diferente al anterior, en la siguiente variable "prevTouchDeltaMag" conseguimos establecer la diferencia total entre la distancia de donde estaban los dedos y donde están ahora.

Finalmente, se hace una resta para ver si el incremento es positivo (Zoom IN) o negativo (Zoom Out) con la variable "delta MagnitudeDiff" que obtiene un único valor con el que actual.

Para imprimir esta orden a la cámara, se usa el comando:

```
GetComponent<Camera>().fieldOfView += deltaMagnitudeDiff * perspectiveZoomSpeed;
```

Que le dice a la cámara donde situarse con respecto al valor obtenido. Para que haya unas delimitaciones y el zoom no sea infinito, se usa el código:

```
GetComponent<Camera>().fieldOfView = Mathf.Clamp(GetComponent<Camera>().fieldOfView, 25f, 60f);
```

que establece unos valores mínimos y máximos de posición de la cámara para que no se pueda acercar demasiado al objeto ni alejar más de la posición inicial.

Luego, en lo que resta de funcionalidad, solo nos queda hablar de como interactuar con los botones. Para ello Unity incorpora unas funciones que nos ayudarán bastante en nuestro proyecto, y nos ahorrarán redactar mucho código. Primero debemos tener un script de funciones, en el que escribiremos como son las ordenes que activamos. Este script, debemos tenerlo presente también en el escenario virtual para asociarlo a los elementos gráficos que vamos a ver relacionados. Primero veamos el código que usaremos:

```
using UnityEngine;
using UnityEngine.SceneManagement;

public class Botones : MonoBehaviour
{
    float valorcam = -8;

    public void EntrarMapa()
    {
        SceneManager.LoadScene("Mapa");
    }

    public void EntrarGuia1()
    {
        SceneManager.LoadScene("Guia1");
    }
    public void EntrarGuia2()
    {
        SceneManager.LoadScene("Guia2");
    }
    public void EntrarGuia3()
    {
        SceneManager.LoadScene("Guia3");
    }
    public void EntrarGuia4()
    {
        SceneManager.LoadScene("Guia4");
    }
    public void EntrarGuia5()
    {
        SceneManager.LoadScene("Guia5");
    }
    public void EntrarGuia6()
    {
        SceneManager.LoadScene("Guia6");
    }

    public void BotonSalir()
    {
        Application.Quit();
    }

    public void EntrarMenu()
    {
        SceneManager.LoadScene("Menu");
    }
}
```

Como podemos ver, la estructura de una función es muy simple, parte de la declaración (`public void nombre`) y entre llaves las ordenes que queremos que ejecuten. Hecho esto, en la escena de Unity, si vamos a un elemento gráfico, le podemos asignar la función de botón, lo que le da la habilidad de incorporar un script que le marque que hacer cuando es pulsado.

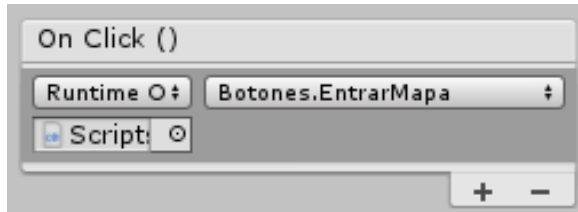


Ilustración 57 - Detalle programa Unity

Podemos ver por el título de la opción que efectuará órdenes cuando sea “pulsado”. Si creamos un objeto vacío en el escenario virtual, lo anexamos al código que hemos mostrado, y este objeto lo ponemos en esta función, nos permite seleccionar a la derecha entre las funciones que se encuentran

en el, en este ejemplo vemos como hemos seleccionado “Botones.EntrarMapa”. Lo que significa que dentro de el script Botones dode redactamos todas las funciones, entra en la que se llama “EntrarMapa” la cual se ejecutará, en el ejemplo, cambiará de escena e iremos a la escena Mapa.

```
public void EntrarMapa()  
{  
    SceneManager.LoadScene("Mapa");  
}
```

Con esta simpleza, podemos programar todos los botones para que realicen la función que queramos, ya sea dirigirnos a otra pantalla (escena) o efectuar una acción concreta como desplegar un texto o activar el audio, por ejemplo. Para esto último, hemos programado un botón de audio dentro de cada una de las escenas que contienen los cráneos en el que se activa la audio-descripción del texto informativo. Esta orden se consigue a través de programar el botón también como “Audio Source”, lo que le confiere la habilidad de anexarle un archivo de audio y reproducirlo. Programando conseguimos que al pulsarlo se reproduzca el audio, y se active el botón de pause, que cuando se pulse, se pausará el audio y desactivará dejando visible otra vez el botón de Audio.

Para los gráficos, el nivel de detalle de los elementos en escena y las luces y sombras que entran en juego, se utiliza un pequeño código que nos permite establecer niveles de detallismo y con ello activar unos elementos u otros. Por ejemplo, en esta aplicación hemos dividido el espectro en tres posibilidades. Viendo que la gente tiene tres tipos de smartphones, alta gama, gama media y gama baja, se podrán adecuar totalmente a ellos. Primeramente, se configuran los botones de configuración gráfica para que activen las órdenes de poner en acción el modo de calidad gráfica que queremos establecer. Lo vemos en el código:

```

using UnityEngine;

public class Graficos : MonoBehaviour
{
    public void GraficosBajos()
    {
        QualitySettings.SetQualityLevel(0, true);
    }
    public void GraficosMedios()
    {
        QualitySettings.SetQualityLevel(1, true);
    }
    public void GraficosAltos()
    {
        QualitySettings.SetQualityLevel(2, true);
    }
}

```

Esto establece los tres niveles, pero ¿Qué propiedades tiene cada nivel gráfico? Lo podemos configurar en una herramienta que nos proporciona Unity en la que seremos capaces de adaptarlo a las necesidades y capacidades de cada gama de smartphones. Con esto podemos tener varios objetos iguales, pero con distinta carga gráfica, es decir menos o mayor número de polígonos, y activarlos dependiendo del nivel gráfico que tengamos

activado. Si por ejemplo estamos en la configuración más baja, se nos mostrarán objetos con menos polígonos, menos detallados, pero si por el contrario estamos en el modo de alta calidad gráfica, activaremos los elementos con mayor resolución y mayor número de polígonos.

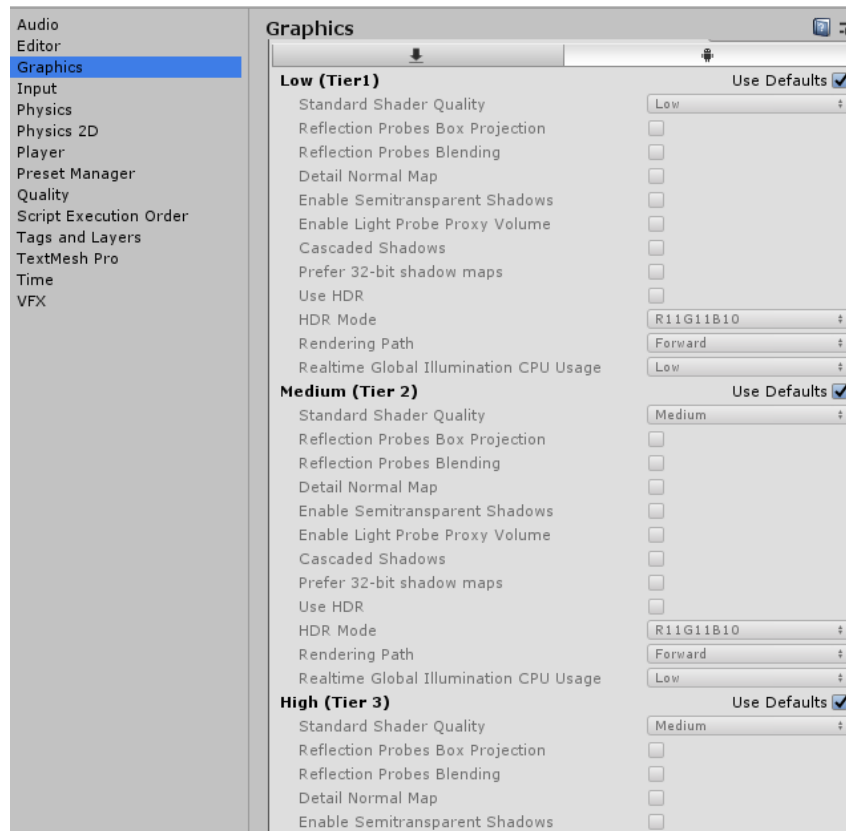


Ilustración 58 - Detalle ajustes gráficos en Unity

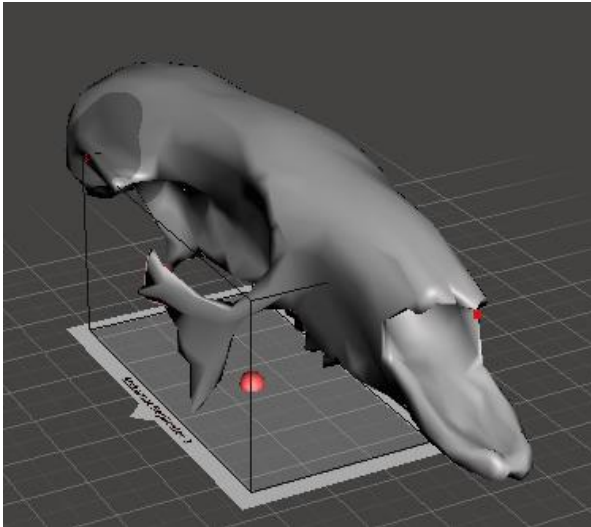


Ilustración 59 - Cráneo con poca definición

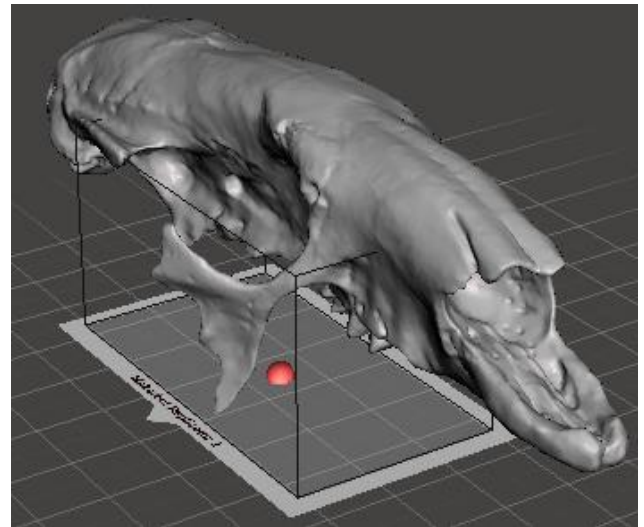


Ilustración 60 - Cráneo con mucha definición

Este es un ejemplo de cráneo en baja resolución (izquierda) con muy pocos polígonos, y uno con alta resolución (derecha) con gran cantidad de polígonos (mas de 100000). Esta es una técnica que podemos hacer fácilmente con la herramienta Meshmixer, la cual seleccionando el objeto, y editando su número de triángulos, conseguimos variar drásticamente su carga gráfica en Unity. Esta es una técnica muy usada para conseguir renderizar multitud de elementos en imagen. Primero se establece la configuración gráfica, ella marca el nivel de detalle de todos los objetos, luego, se calcula la distancia hasta el observador (la cámara en nuestro caso) y finalmente se carga el objeto con mayor o menor definición dependiendo de la distancia, y la importancia del mismo.

10. Descripción detallada de la solución adoptada:

El resultado final lo vamos a presentar con diferentes capturas de las pantallas que presenta la aplicación. Para ello, iremos de las que primero podremos encontrar a las últimas.

- **Pantalla inicial, el menú:** En él podemos ver a primera vista los colaboradores de la aplicación, como son la Universitat Politècnica de Valencia, y el Ajuntament de Valencia. Luego, tres opciones que nos indican “INICIAR”, “CONFIGURACIÓN GRÁFICA” e “INFORMACIÓN”. Además de un botón para salir. Todo ello acompañado por un símbolo gráfico que ayuda visualmente a entender para que sirve cada opción sin necesidad de leer casi.

El fondo de la pantalla es una visión en rotación continua en la que podemos ver diferentes dinosaurios en una selva flanqueada por montañas. Esto fue ideado para dar una sensación fuerte al principio al usuario de estar ya interactuando con dinosaurios y que quiera adentrarse aun más en la aplicación.



Ilustración 62 - Pantalla inicial/Menú

- **Pantalla para la Configuración Gráfica:** Donde el usuario selecciona el modo de carga gráfica que su terminal puede ejecutar.

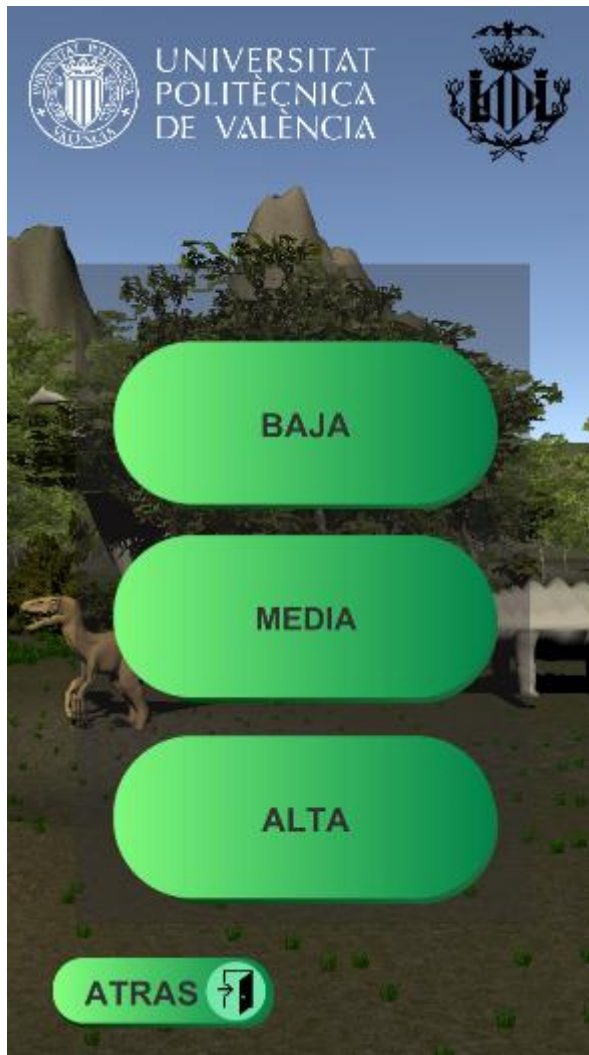


Ilustración 63 - Pantalla Configuración Gráfica

- **Pantalla de Información:** Con una breve explicación de que es la aplicación y por qué se hizo.



Ilustración 64 - Pantalla información

- **Pantalla de Selección:** Donde el usuario se va desplazando entre dinosaurios para seleccionar el que desea. Incorpora un título que indica que hacer en este momento (“¡Elige Dinosaurio!”) así como el nombre del mismo y la imagen que sirve de enlace para entrar al contenido. En el fondo de

la pantalla vemos una fotografía general de la exposición Rodrigo Botet.

- **Pantalla del contenido:** En la que ya podemos interactuar con el fósil, y rotarlo. De fondo la imagen del dinosaurio y en la parte inferior los botones de Salir y Información. Este último lo veremos a continuación.

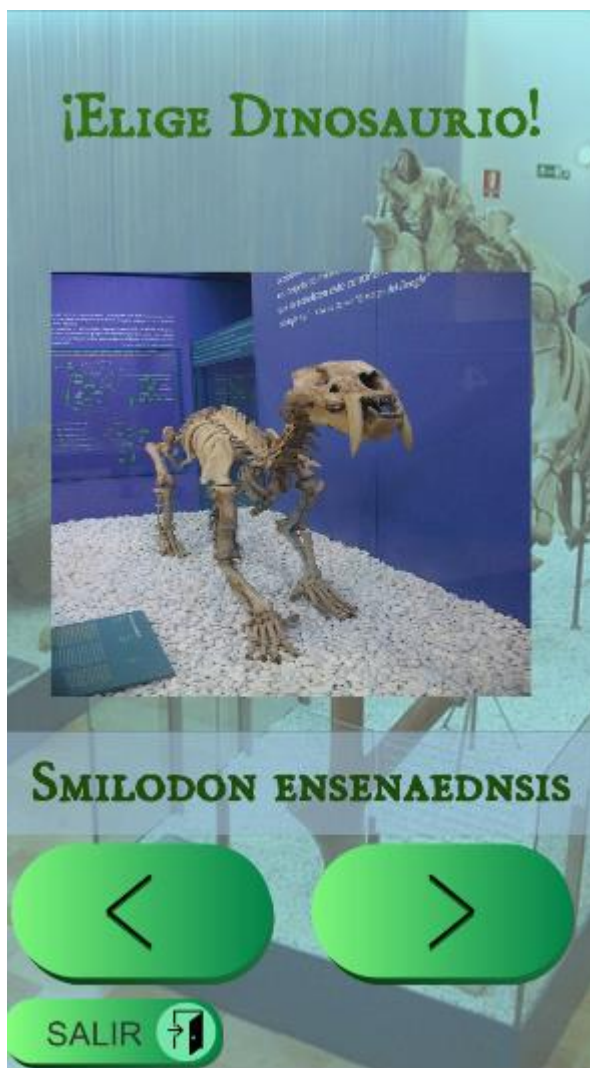


Ilustración 65 - Pantalla selección



Ilustración 66 - Pantalla de Interacción

- **Pantalla de Información:** Se superpone un recuadro con el nombre del dinosaurio y con el texto informativo. Aquí dentro se encuentra la opción de activar el audio. El texto se desliza con una barra de scroll a la izquierda o tocando y desplazando el propio texto.

Scelidotherium Carlesi:

Pertenece al grupo de los perezosos gigantes. La estructura de su cráneo y sus dientes indica una alimentación básicamente herbívora.

Procedente de las "Toscas del Río de La Plata". Scelidotherium se registra en el Ensenadense de Buenos Aires y en el Pleistoceno de San Luis. La mayoría de los especímenes procedentes de las "toscas del Río de La Plata" corresponden a elementos aislados del post-cráneo. Las primeras descripciones fueron realizadas por Charles Darwin en Bahía Blanca, Argentina en su viaje del Beagle. Basados en cráneos, vértebras y



Ilustración 67 - Pantalla de Información adicional

11. Alternativas propuestas

Cuando se proponen ideas para la propuesta cromática, también se generan ideas acerca de darle un enfoque más futurista y un empaque más moderno, cambiando el tono verdoso por el azulado en los botones, pero se acabó descartando, aunque es una gran alternativa.

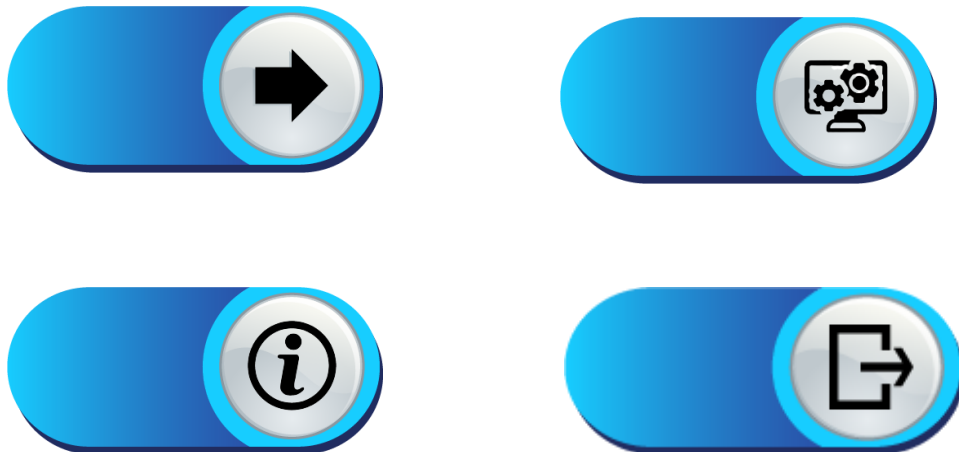


Ilustración 68 - Botones azules / Alternativa visual

Otra alternativa planteada es el desarrollo de la aplicación para un visionado del dispositivo en horizontal, lo que facilitaría un poco la colocación de los botones y de la visión del propio cráneo, pero esta versión fue descartada por que obliga al usuario a hacer uso de las dos manos, y los visitantes de un museo casi siempre van con una mano en la mochila o sujetando una botella de agua, o sujetando un souvenir o algo por el estilo. Aunque en vertical se pierde un poco en disposición y en comodidad visual, se gana en comodidad ergonómica.

Una alternativa originada al principio fue que los botones fueran más delgados, pero fue declinada porque mucha gente que visita los museos son gente mayor que tiene vista reducida y el gran tamaño de los botones les ayuda a la fácil interacción con la aplicación.

12. Conclusiones

Para concluir, cabe destacar el logro de haber resuelto el objetivo principal, generando una aplicación que cumple con todos los requisitos que se planteaban y que, en la puesta a prueba con la gente, le gusta.

El proyecto a sido llevado a cabo con gran entusiasmo y ganas de aprender a usar un software nuevo, poner a prueba un gran proyecto como este y probar las dificultades reales que en este camino se encuentran.

La complejidad a la hora de desarrollar una aplicación viene de muchos aspectos a la vez, la dificultad técnica cuando se usan softwares informáticos, complejidad del entorno, porque tienes que comprender y regular una multitud de parámetros que influyen en el diseño del proyecto y dificultad de aprendizaje, ya que de plantea un trabajo en algo que no se ha especializado el alumno en el transcurso del grado y por lo tanto es un “mundo nuevo” para él.

Unos de los puntos más importantes en el momento en que se termina este trabajo, es plantearse muchas mejoras, que antes, por la inexperiencia del alumno no eran planteadas y que ahora, con la práctica de este proyecto, si son asumibles para este como para otros futuros proyectos. A sido todo un reto aprender y mejorar cada día pero el resultado final hace pensar que se han conseguido aquellos objetivos de los que se habla al principio del trabajo, como de los que no, que son aprender y especializarse en una rama del diseño que tiene mucho futuro, la digitalización 3D y las aplicaciones para smartphones.

13.Fuentes de Información:

Webgrafía:

<http://play.google.com>

<https://unity.com/es>

<https://blogthinkbig.com/motores-graficos>

<https://www.hobbyconsolas.com/listas/mejores-motores-graficos-hacer-tus-propios-juegos-302029>

<http://es.wikipedia.org>

<https://www.vidaextra.com/listas/si-quieres-hacer-tus-propios-juegos-estos-son-los-mejores-motores-que-vas-a-encontrar>

<https://www.3dnatives.com/es/top-10-escaneres-3d-low-cost-19102016/>

<https://www.dafont.com/es/>

<http://www.meshmixer.com/>

<https://www.blender.org/>